Assignment 5

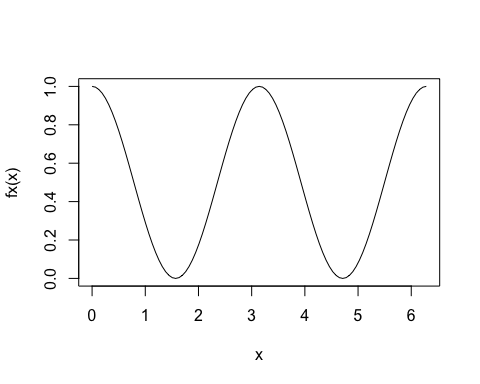
Wenhao Li 250959325

30/03/2020

# Question 1

# (a)

fx<-function(x){  
 cos(x)^2  
 return(cos(x)^2)  
}  
curve(fx,from=0,to=2\*pi)



# (b)

set.seed(721)  
n<-1000000  
a<-runif(n,min=0,max=2\*pi)  
estimator<-mean(fx(a)/dunif(a,min=0,max=2\*pi))  
variance<-(mean((fx(a)/dunif(a,min=0,max=2\*pi))^2)-estimator^2)/n  
b<-sqrt(variance)  
d<-estimator-1.96\*b  
e<-estimator+1.96\*b  
c(d,e)

## [1] 3.140547 3.149255

estimator

## [1] 3.144901

Comment:

The area is 3.144901 The confidence interval is (3.140547,3.149255)

# (c)

电脑屏幕的照片上有文字

描述已自动生成

confidence.interval<-c(d,e)  
pi<3.149255

## [1] TRUE

3.140547<pi

## [1] TRUE

Comment:

The exactly area is pi. The confidence interval covers the true value.

# Question 2

# (a)

set.seed(721)  
x<-rnorm(100,10,2)  
mean(x)

## [1] 9.784903

sd(x)

## [1] 2.299072

min(x)

## [1] 3.930008

max(x)

## [1] 13.98267

Comment:

Its mean is 9.784903.

The standard deviation is 2.299072.

The minimum value is 3.930008.

The maximum value is 13.98267.

# (b)

set.seed(721)  
booted.data<-replicate(50000,mean(sample(x,replace=TRUE)))  
mean(booted.data)

## [1] 9.784473

sd(booted.data)

## [1] 0.2290073

min(booted.data)

## [1] 8.69461

max(booted.data)

## [1] 10.72199

Comment:

Its mean is 9.784473.

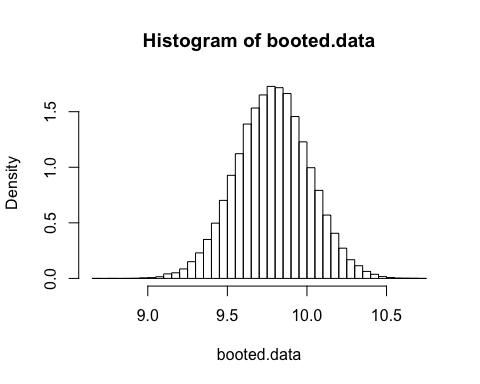
The standard deviation is 0.2290073.

The minimum value is 8.69461.

The maximum value is 10.72199.

# (c)

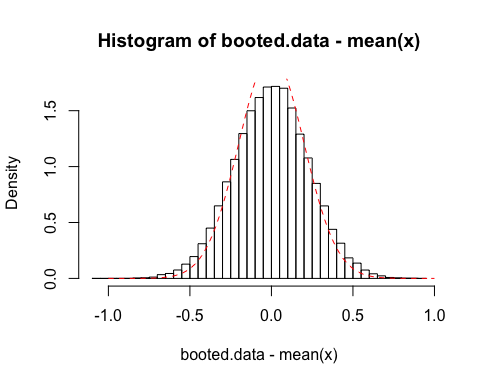
hist(booted.data,breaks=36,probability=TRUE)

 The shape of this distribution is close to a normal distribution.

The center of this distribution is around 9.8.

# (d)

theoretical.density<-function(x){  
 dnorm(x,mean=0,sd=2/10)  
}  
hist(booted.data-mean(x),probability=TRUE,breaks=30)  
curve(theoretical.density,add=TRUE,lty=2,col=2,from=-1,to=1)

 Comment:

The shape is close to a normal distribution with center at 0.

The histogram of booted.data-mean(x) seems to fit the theoretical density (plot).

# (e)

# Repeat first time  
set.seed(720)  
x<-rnorm(100,10,2)  
mean(x)

## [1] 9.405577

sd(x)

## [1] 2.054628

min(x)

## [1] 3.581669

max(x)

## [1] 13.67049

set.seed(720)  
booted.data<-replicate(50000,mean(sample(x,replace=TRUE)))  
mean(booted.data)

## [1] 9.406519

sd(booted.data)

## [1] 0.2042293

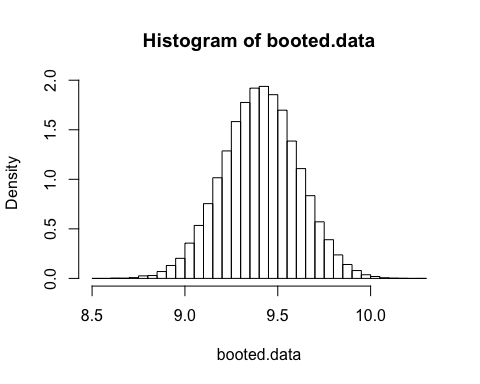
min(booted.data)

## [1] 8.52795

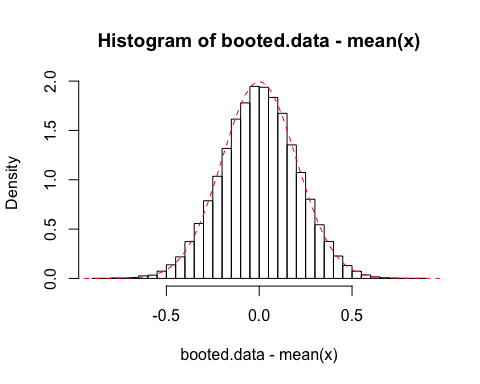
max(booted.data)

## [1] 10.28952

hist(booted.data,breaks=36,probability=TRUE)



theoretical.density<-function(x){  
 dnorm(x,mean=0,sd=2/10)  
}  
hist(booted.data-mean(x),probability=TRUE,breaks=30)  
curve(theoretical.density,add=TRUE,lty=2,col=2,from=-1,to=1)



# Repeat second time  
set.seed(722)  
x<-rnorm(100,10,2)  
mean(x)

## [1] 9.990996

sd(x)

## [1] 1.792794

min(x)

## [1] 6.249871

max(x)

## [1] 14.86082

set.seed(722)  
booted.data<-replicate(50000,mean(sample(x,replace=TRUE)))  
mean(booted.data)

## [1] 9.992565

sd(booted.data)

## [1] 0.1782085

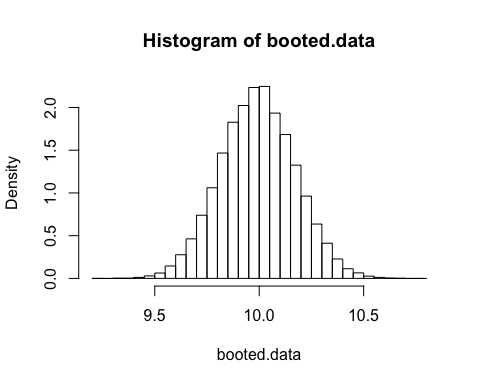
min(booted.data)

## [1] 9.209986

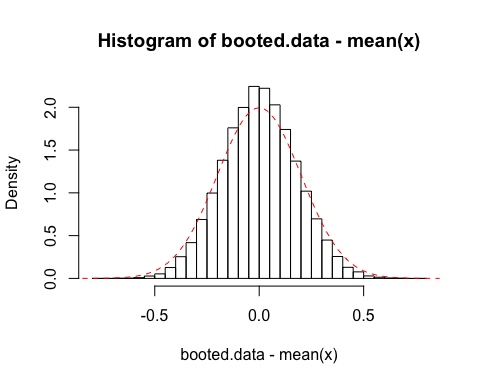
max(booted.data)

## [1] 10.76077

hist(booted.data,breaks=36,probability=TRUE)



theoretical.density<-function(x){  
 dnorm(x,mean=0,sd=2/10)  
}  
hist(booted.data-mean(x),probability=TRUE,breaks=30)  
curve(theoretical.density,add=TRUE,lty=2,col=2,from=-1,to=1)

 Comment:

The result changes a little.

However, the shape of both new distributions are consistent with previous distribution.

They are all close to normal distribution (bell shape).

# Question 3

# (a)

my.obj<-function(theta,x){  
 a<-sum(abs(x-theta))  
 return(a)  
}

# (b)

optimize.b<-function(x,interval=c(min(x),max(x))){  
 b<-optimize(my.obj,x=x,interval=interval)  
 return(b$minimum)  
}  
# Test   
optimize.b(c(3,7,9,12,15,18,21))

## [1] 12

# (c)

nlminb.c<-function(x,start=mean(x)){  
 c<-nlminb(start,objective=my.obj,x=x)  
 return(c$par)  
}  
# Test  
nlminb.c(c(3,7,9,12,15,18,21))

## [1] 12

# (d)

k<-c(1,3,7,9,12,15,18,21)  
optimize.b(k,interval=c(0,22))

## [1] 10.37631

optimize.b(k,interval=c(0,32))

## [1] 10.64031

optimize.b(k,interval=c(0,40))

## [1] 9.442769

nlminb.c(k,start=7)

## [1] 12

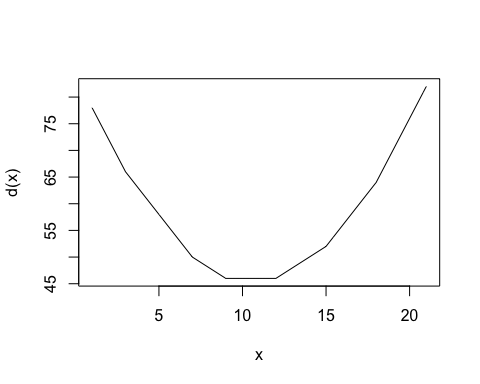
nlminb.c(k,start=21)

## [1] 9

nlminb.c(k,start=30)

## [1] 10

d<-function(theta.d){  
 sapply(theta.d,function(theta)my.obj(theta,k))  
}  
curve(d,from=min(k),to=max(k))

 Comment:

With three different wider intervals, function from (b) has different results.

With three different start vaules,function from (c) has different results.

The graph fits the above values.

When x is from 9 to 12, it has minimum value.

# Question 4

# (a)

"huron" <-   
structure(.Data = c(581.55999999999995, 581.54999999999995, 581.34000000000003, 580.84000000000003,   
 580.33000000000004, 580.35000000000002, 579.87, 580.49000000000001,   
 579.90999999999997, 580.07000000000005, 580.90999999999997, 581.10000000000002,  
 579.72000000000003, 580.32000000000005, 580.48000000000002, 580.38,   
 581.86000000000001, 580.97000000000003, 580.79999999999995, 579.78999999999996,  
 580.38999999999999, 580.41999999999996, 580.82000000000005, 581.39999999999998,  
 581.32000000000005, 581.44000000000005, 581.67999999999995, 581.16999999999996,  
 580.52999999999997, 580.00999999999999, 579.90999999999997, 579.13999999999999,  
 579.15999999999997, 579.54999999999995, 579.66999999999996, 578.44000000000005,  
 578.24000000000001, 579.10000000000002, 579.09000000000003, 579.35000000000002,  
 578.82000000000005, 579.32000000000005, 579.00999999999999, 579.,   
 579.79999999999995, 579.83000000000004, 579.72000000000003, 579.88999999999999,  
 580.00999999999999, 579.37, 578.69000000000005, 578.19000000000005,   
 578.66999999999996, 579.54999999999995, 578.91999999999996, 578.09000000000003,  
 579.37, 580.13, 580.13999999999999, 579.50999999999999, 579.24000000000001,   
 578.65999999999997, 578.86000000000001, 578.04999999999995, 577.78999999999996,  
 576.75, 576.75, 577.82000000000005, 578.63999999999999, 580.58000000000004,   
 579.48000000000002, 577.38, 576.89999999999998, 576.94000000000005,   
 576.24000000000001, 576.84000000000003, 576.85000000000002, 576.89999999999998,  
 577.78999999999996, 578.17999999999995, 577.50999999999999, 577.23000000000002,  
 578.41999999999996, 579.61000000000001, 579.04999999999995, 579.25999999999999,  
 579.22000000000003, 579.38, 579.10000000000002, 577.95000000000005, 578.12, 579.75,  
 580.85000000000002, 580.40999999999997, 579.96000000000004, 579.61000000000001,  
 578.75999999999999, 578.17999999999995, 577.21000000000004, 577.13,   
 579.10000000000002, 578.25, 577.90999999999997, 576.88999999999999,   
 575.96000000000004, 576.79999999999995, 577.67999999999995, 578.38,   
 578.51999999999998, 579.74000000000001, 579.30999999999995, 579.88999999999999,  
 579.96000000000004, 580.98000000000002, 581.03999999999996, 580.49000000000001,  
 580.51999999999998, 578.57000000000005, 578.96000000000004, 579.94000000000005,  
 579.76999999999998, 579.44000000000005, 578.97000000000003, 580.08000000000004,  
 580.23000000000002, 580.75, 581.26999999999998)  
, class = c("ts", "numeric")  
, tsp = c(1860., 1986., 1.)  
, title = "Lake Huron, mean level, July, 1860-1986"  
)  
mean(huron)

## [1] 579.3091

sd(huron)

## [1] 1.335657

min(huron)

## [1] 575.96

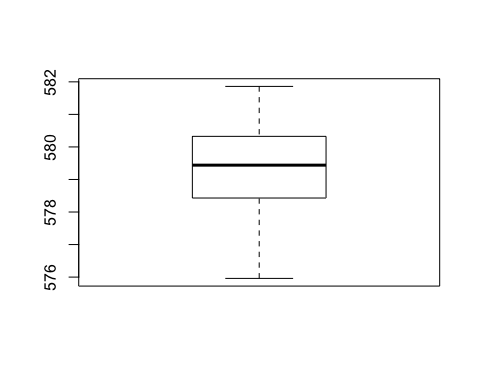
max(huron)

## [1] 581.86

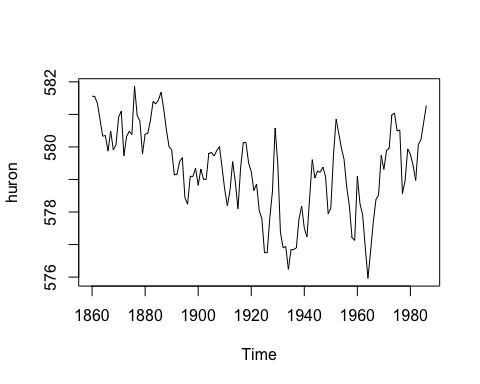
median(huron)

## [1] 579.44

boxplot(huron)



plot.ts(huron)

 Comment:

The mean value is 579.3091.

The standard deviation value is 1.335657.

The min value is 575.96.

The max value is 581.86.

The median value is 579.44.

The boxplot shows that it does not have outliers and it seems to have a little left skewness.

The time series shows that it changes all the time.

The mean value becomes smaller as times goes, especially the mean value from 1900 to 1980 is lower than the mean valur from 1860 to 1890.

The variance becomes larger as time goes.

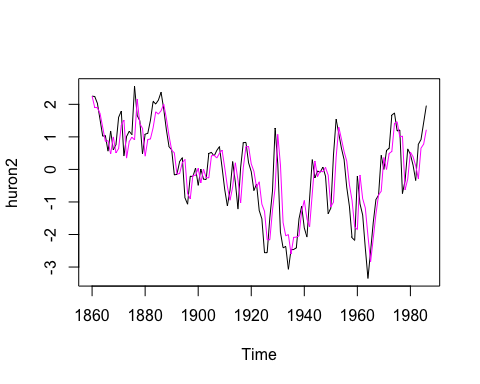
# (b)

log.likelihood=function(par, x){  
n=length(x)  
v=x[1]^2  
for (i in 2:n)  
v=v+(x[i]-par[1]\*x[i-1])^2  
return(v/par[2]+n\*log(par[2]))  
}  
huron2<-huron-mean(huron)  
a<-nlminb(start=c(1,1),objective=log.likelihood,x=huron2,lower=c(-21,10^(-7)),upper=c(21,70))  
b<-a$par[1]  
b

## [1] 0.8459369

# (c)

pred.huron2=huron2  
for (i in 2:length(huron2)) {  
 pred.huron2[i]=b\*huron2[i-1]  
}  
plot(huron2)  
lines(pred.huron2,col=22)

 Comment:

As we seen from the plot, at the same time (year), there is difference between true values and predicted values.

So, the true values’ distribution and the predicted values’ distribution are different. However, the overall changing pattern of two distribution is nearly same.